

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Appellant:	Ramamoorthy	Patent Application	
Serial No.:	10/723,643	Group Art Unit:	2441
Filed:	November 24, 2003	Examiner:	Patel, Chirag

For: DYNAMICALLY BALANCING LOAD FOR SERVERS

Appeal Brief

Table of Contents

	<u>Page</u>
Real Party in Interest	2
Related Appeals and Interferences	3
Status of Claims	4
Status of Amendments	5
Summary of Claimed Subject Matter	6
Grounds of Rejection to be Reviewed on Appeal	8
Arguments	9
Claims Appendix	14
Evidence Appendix	19
Related Proceedings Appendix	20

Real Party in Interest

The assignee of the present invention is Hewlett-Packard Development Company, L.P., a Texas Limited Partnership.

Related Appeals and Interferences

There are no related appeals or interferences known to the Appellant.

Status of Claims

Claims 1-20 are pending. Claims 1-20 stand rejected. Rejections of Claims 1-20 are herein appealed.

Status of Amendments

All proposed amendments have been entered. An amendment subsequent to the Final Action has not been filed.

Summary of Claimed Subject Matter

Independent Claim 1 recites a method (300 of Figure 3 and page 9, line 15) of dynamically balancing load in a system of servers including monitoring (310 of Figure 3 and page 9, line 18) for servers that are able to respond to requests directed at the system, including actively discovering new servers in the system of servers. The method also includes determining (315 of Figure 3 and page 10, line 5) a performance metric for a first set of the servers discovered by the monitoring for the servers, maintaining (340 of Figure 3 and page 12, line 7) a table comprising the performance metric for the first set of discovered servers and in response to receiving a request, routing (430 of Figure 4 and page 13, line 6) the request to a selected server in the system of servers based on the performance metric, wherein the system of servers comprises the first set of discovered servers.

Independent Claim 8 recites a method of dynamically balancing load. The method includes dynamically discovering (310 of Figure 3 and page 9, line 18) a first set of servers that are able to respond to requests directed at a system and actively discovering new servers of the system, determining (315 of Figure 3 and page 10, line 5) a response time of each of the first set of discovered servers, calculating stress factors (315 of Figure 3 and page 10, line 5) for each of the first set of discovered servers, based in part on the response time, receiving a request to the system (410 of Figure 4 and page 12, line 20), determining (420 of Figure 4 and page 12, line 24) a server in the system to route the request to based on the stress factors, wherein the system comprises the first set of discovered servers and routing (430 of Figure 4 and page 13, line 6) the request to said server in the system determined in said determining (420 of Figure 4 and page 12, line 24).

Independent Claim 14 recites a system (150 of Figure 1 and page 5, line 24) for balancing load. The system includes a plurality of back-end servers (180a-180n of Figure 1 and page 6, lines 1-13) that are able to service requests to the system, a front-end server (160a of Figure 1 and page 6, lines 1-13) having a load balancing agent (170a of Figure 1 and page 6, lines 15-25) comprising a table (200a of Figure 1 and page 8, lines 1-18), wherein the front-end server receives requests that are forwarded to the back-end servers, and wherein the load balancing agent is operable to monitor (310 of Figure 3 and page 9, line 18) for back-end servers that are able to service requests to the system, including actively discovering new back-end servers, determine (315 of Figure 3 and page 10, line 5) a performance metric for the back-end servers discovered by the monitoring and determine (420 of Figure 4 and page 12, line 24) a server of the back-end servers to route a request to based on the performance metric.

Grounds of Rejection to be Reviewed on Appeal

1. Claims 1-20 stand rejected under 35 U.S.C. 103(a) as being obvious over Kazemi (7,089,281) in view of Hickman (6,523,026).

Arguments

1. Whether Claims 1-20 stand rejected under 35 U.S.C. 103(a) as being obvious over Kazemi (7,089,281) in view of Hickman (6,523,026).

“As reiterated by the Supreme Court in *KSR*, the framework for the objective analysis for determining obviousness under 35 U.S.C. 103 is stated in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966). Obviousness is a question of law based on underlying factual inquiries” including “[a]scertaining the differences between the claimed invention and the prior art” (MPEP 2141(II)). “In determining the differences between the prior art and the claims, the question under 35 U.S.C. 103 is not whether the differences themselves would have been obvious, but whether the claimed invention as a whole would have been obvious” (emphasis in original; MPEP 2141.02(I)). Appellants note that “[t]he prior art reference (or references when combined) need not teach or suggest all the claim limitations, however, Office personnel must explain why the difference(s) between the prior art and the claimed invention would have been obvious to one of ordinary skill in the art” (emphasis added; MPEP 2141(III)).

Appellants submit that Kazemi fails to teach each element of Independent Claim 1. Specifically, Appellant submits that Kazemi fails to teach or suggest (emphasis added) “monitoring for servers that are able to respond to requests directed at the system, including actively discovering new servers in said system of servers,” as claimed. Hickman also fails to teach or suggest this claimed feature.

Appellants also submit that Kazemi fails to teach or suggest “determining a performance metric for the first set of servers discovered by said monitoring,” as claimed. Hickman also fails to teach or suggest this claimed feature.

Kazemi is used for “storage resources” as opposed to server resources as with the present invention. Appellants would like to point out that there is a distinct difference between the two. For example, in a storage resource system, data that is stored on a server “a” can not be accessed from server “b” because the information is stored on server “a”. In a storage system redundancy is used to preserve data in case of failure. With a server resource environment, a request to perform an operation on server “a” can be routed to server “b” as long as server “b” has the required performance metric to respond to the request.

Appellant respectfully submits that Kazemi fails to teach or suggest “monitoring for servers that are able to respond to requests directed at the system including actively discovering new servers in said system of servers,” as claimed. In opposition to “monitoring” and “actively discovering” Kazemi relies on a passive system wherein the data storage units report available resources. Appellants submit that receiving reports is very different from actively discovering, as claimed.

In column 11, lines 26-31, Kazemi states “populating the resource table as information is received from servers 210 as they come online.” With Kazemi, the DSR relies on the servers themselves to report performance information. The passive DSR of Kazemi merely receives data from the servers and does not monitor or actively discover, as claimed.

Kazemi uses a DSR (dynamic session redirector) to provide a single system image for resources that are distributed on many different servers (summary of the invention). “The DSR examines the incoming requests and passes them on to the appropriate resources and then forwards the results of any request to the requesting client.” Kazemi teaches “DSR keeps a table of the resources provided by each network storage device.” With Kazemi, the locations of resources are known and kept track of in a table. Kazemi fails to teach or suggest discovery of any kind.

In addition, Appellants submit that Kazemi fails to teach or suggest “determining a performance metric for the first set of servers discovered by said monitoring,” as claimed. With the present invention, monitoring is used to discover servers capable of responding to a requests including actively discovering new servers in the system of servers.

Performance metrics are then determined for the discovered servers. Appellants submit that Kazemi does not teach or suggest “discovering” servers. With Kazemi, the particular resources and associated servers are known and not discovered since Kazemi is dealing with a storage system and must know where the data is located in order to retrieve it. Furthermore, with Kazemi, the DSR requires the servers themselves to report the performance information which is clearly different from discovering, as claimed.

It is necessary for Kazemi to know where data is stored so the data can be retrieved since Kazemi deals with data storage systems. If Kazemi had to discover where the data was stored, the response time would be severely impacted.

Appellants submit that discovery would not be an obvious modification to Kazemi because Kazemi is directed to storage servers where discovery would significantly decrease performance. Appellants have reviewed Hickman and submit that Hickman fails to remedy the deficiencies of Kazemi. In particular, Appellants submit that Hickman also fails to teach monitoring to discover servers, including actively discovering new servers in the system of servers, as claimed.

For this rational, Claims 1-20 are patentable over Kazemi in view of Hickman. As such, Appellants respectfully request the rejection be removed.

In summary, the Appellant respectfully requests that the Board reverse the Examiner's rejections of claims 1-20.

The Appellant wishes to encourage the Examiner or a member of the Board of Patent Appeals to telephone the Appellant's undersigned representative if it is felt that a telephone conference could expedite prosecution.

Respectfully submitted,

WAGNERBLECHER LLP

Date: 03/26/2009

/John P. Wagner, Jr./

John P. Wagner, Jr.

Registration Number: 35,398

WAGNERBLECHER
WESTRIDGE BUSINESS PARK
123 WESTRIDGE DRIVE
WATSONVILLE, CALIFORNIA 95076
408-377-0500

Claims Appendix

1. A method of dynamically balancing load in a system of servers, comprising:

a) monitoring for servers that are able to respond to requests directed at the system, including actively discovering new servers in said system of servers;

b) determining a performance metric for a first set of said servers discovered by said monitoring for the servers;

c) maintaining a table comprising said performance metric for said first set of discovered servers; and

d) in response to receiving a request, routing said request to a selected server in the system of servers based on said performance metric, wherein the system of servers comprises the first set of discovered servers.

2. The method of Claim 1, further comprising:

determining a load on ones of the servers in the system of servers.

3. The method of Claim 2, further comprising:

determining a stress factor for a given server based on the performance metric of the given server and the load on the given server.

4. The method of Claim 1, further comprising:

200208157-1

Serial No.: 10/723,643
Group Art Unit: 2441

determining a stress factor for ones of the servers in the system of servers based on the performance metrics.

5. The method of Claim 1, wherein the performance metric is a response time.

6. The method of Claim 1, wherein the performance metric is a response time when the servers discovered by said monitoring are unloaded.

7. The method of Claim 1, further comprising:

periodically reevaluating said performance metric for the servers in the system of servers.

8. A method of dynamically balancing load, comprising:

a) dynamically discovering a first set of servers that are able to respond to requests directed at a system and actively discovering new servers of said system;

b) determining a response time of each of the first set of discovered servers;

c) calculating stress factors for each of the first set of discovered servers, based in part on said response time;

d) receiving a request to the system;

e) determining a server in the system to route the request to based on the stress factors, wherein the system comprises the first set of discovered servers;
and

f) routing said request to said server in the system determined in said e).

9. The method of Claim 8, wherein said b) comprises determining a response time for each of the first set of discovered servers to a request.

10. The method of Claim 8, wherein said b) comprises determining a response time for each of the first set of discovered servers to a database query.

11. The method of Claim 8, wherein said c) comprises calculating the stress factor for each of the first set of discovered servers, based on said response time and a load for each of the first set of discovered servers.

12. The method of Claim 8, wherein:

said b) further comprises determining a response time of a second set of discovered servers not discovered in said a);

said c) comprises calculating stress factors for each of the second set of discovered servers not discovered in said a), wherein the system further comprises the second set of discovered servers not discovered in said a).

13. The method of Claim 8, wherein said second set of discovered servers not discovered in said a) are reported to a load-balancing agent in a configuration file.

14. A system for balancing load, comprising:

a plurality of back-end servers that are able to service requests to the system;

a front-end server having a load balancing agent comprising a table, wherein said front-end server receives requests that are forwarded to said back-end servers, and wherein said load balancing agent is operable to:

monitor for back-end servers that are able to service requests to the system, including actively discovering new back-end servers;

determine a performance metric for the back-end servers discovered by the monitoring; and

determine a server of said back-end servers to route a request to based on the performance metric.

15. The system of Claim 14, wherein said load balancing agent is further operable to determine a load on a given back-end server.

16. The system of Claim 14, wherein said load balancing agent is further operable to determine a stress factor for ones of the back-end servers.

17. The system of Claim 16, wherein the stress factor for a given one of the back-end servers is based on the performance metric and the load on a given one of the given one of the back-end servers.

18. The system of Claim 17, wherein said load balancing agent is able to determine which server of said back-end servers to route a request to based on the stress factor.

19. The system of Claim 14, wherein the performance metric is a response time.

20. The system of Claim 17, wherein said load balancing agent is able to include back-end servers that the load balancing agent did not discover in the determination of which server to route the request to.

Evidence Appendix

None

Related Proceedings Appendix

None